HIPPI-6400 technology dissemination

James Hoffman

Los Alamos National Laboratory, CIC-5, MS-B255
Los Alamos, New Mexico 87545

COPYRIGHT

ABSTRACT

This report covers the technological aspects of High Performance Parallel Interface-6400 (HIPPI-6400), a forthcoming upgrade to the existing HIPPI protocol suite. The report concentrates on the technological advancements and situations that occur in a 6400 Megabit/s network and the solutions produced by the ANSI X3T11 committee.

The first section of the report introduces HIPPI-6400 to familiarize the reader with basic concepts and lay groundwork for future sections. Section 2 analyzes the transmission control link-layer embedded in HIPPI-6400 hardware. Section 3 describes the Scheduled Transfer protocol and connection setup that allows the user to partake of the entire 6400 Mbit bandwidth; followed by Section 4 that describes the signaling interface. Section 5 provides an overview of switching requirements and constructs. Finally, Section 6 concludes the paper, describes works in progress, including simulation, and points to further reading.

Keywords: HIPPI, scheduled transfer, parallel optics, network interface, gigabit network

1. HIPPI-6400 OVERVIEW

The HIPPI-6400-PH standard specifies a point-to-point, duplex, flow-controlled transmission at 6400 Mbit/s of data per direction. Both a parallel copper and fiber interface are planned. Unlike previous HIPPI link levels, HIPPI-6400 uses a 32-byte micropacket to provide an excellent connection setup message structure as well as a building block for large transfers. HIPPI-6400 uses four hardware virtual channel buffers to multiplex between the various message sizes and prevent "jumbogram blockage".

The growing computer and information-sharing technology has created a need for high throughput network communication. The supercomputer interconnect infrastructure tops this technological need by requiring phenomenal bandwidth, gigabytes per second, to support scientific applications. The High Performance Parallel Interface (HIPPI) stands as the current leader for interconnecting high throughput computing and storage nodes. HIPPI has many strengths (maximum 1600 Mbit/s per direction duplex point-to-point links using a “simple” interface) and a few weaknesses (lacking network administration and single-minded connections). The ANSI X3T11 working group is standardizing HIPPI-6400, an upgrade specification that adds increased, end-point applicable bandwidth, lower latency, bandwidth sharing, and greater reliability.

Supercomputer applications require high bandwidth, low latency, reliability, and the capability to handle massive file sizes efficiently. HIPPI-6400 strives to fulfill these needs by fully specifying a 6400 Mbit/s per direction, duplex, point-to-point network interface. HIPPI-6400 uses fixed sized micropackets of 32-data-bytes and 64 bits of control sideband information.

The small micropackets provide a low-latency structure for small messages and a building block for large messages. HIPPI-6400 insures reliable data delivery by using flow-controlled, acknowledged transmission channels and employs two 16-bit cyclic redundancy checks (CRC) to ensure data integrity. To satisfy future bandwidth sharing demands, HIPPI-6400 incorporates four hardware-based virtual channel (VC) buffers that allow multiple outstanding messages, (i.e., large messages no longer block small messages).

Large file transfer (>100 Mbytes) cannot be efficiently transmitted without a scheduled transfer handshake. HIPPI-6400 specifies an platform-independent interoperable scheduled transfer operation set that gives the data receiver time to pin down buffers and specify an exact buffer tiling between the two end points for a smooth transfer placing the bottleneck at the DMA engine I/O-bandwidth (not the operating system response or network protocol side.) Although an old concept, the scheduled transfer has yet to be integrated at such a low layer as in HIPPI-6400.

HIPPI-6400 employs a 4b/5b encoding scheme for balanced transmission. The transmission media is 23 differential-pair copper per direction for 50 meter runs and a 12-wide parallel optical interface per direction for 250 meter runs. The 10 ns skew compensation is required at the receiver end to account for pair to pair skew. The protocol allows 1 km runs, which should be possible by the year 2000 as optics technology improves.

2. TRANSMISSION LINK LAYER

Transmissions in the gigabyte per second range require that error control and flow control be placed into the hardware levels of the networking stack. This allows for a finer granularity of retransmission and buffering than a software based protocol could attend. To provide ultra low latency HIPPI-6400 uses 32-data-byte micropackets, which places a greater need for a hardware error control scheme. To prevent the "busy line" experience seen in legacy HIPPI, a hardware virtual channel mechanism has been implemented with separate credit based flow control for each of the four VC buffers. Flow control and error control have been decoupled to increase performance and decrease complexity. Figure 1 below shows the basic transmission model (Figure 1 comes from HIPPI-6400-PH, the physical layer interface specification). Although all HIPPI-6400 links are duplex, the figure portrays a simplex link to ease understanding (return information is carried in the control bits of micropackets moving in the reverse direction). The transmitting and receiving ports are referred to as Source and Destination respectively. HIPPI-6400 control bits also contain a TYPE field for determining whether is a null micropacket (to keep the link active), a credit-only micropacket, a header micropacket, or a data micropacket.

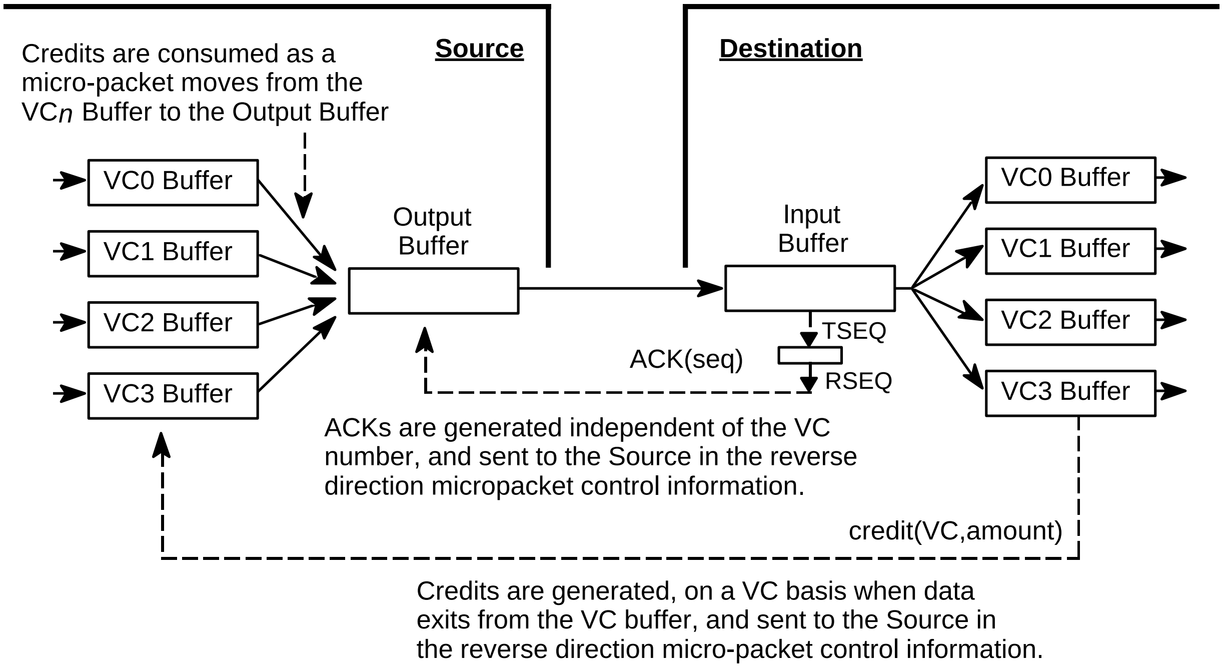

Credits are consumed as a micro-packet moves from the VC*n* Buffer to the Output Buffer
Source
Destination
VC0 Buffer
VC1 Buffer
VC2 Buffer
VC3 Buffer
Output Buffer
Input Buffer
TSEQ
ACK(seq)
RSEQ
VC0 Buffer
VC1 Buffer
VC2 Buffer
VC3 Buffer
ACKs are generated independent of the VC number, and sent to the Source in the reverse direction micropacket control information.
credit(VC,amount)
Credits are generated, on a VC basis when data exits from the VC buffer, and sent to the Source in the reverse direction micro-packet control information.

Figure 1: Transmission Structure

2.1 Error Control and Sequencing

Two 16-bit cyclic redundancy checks (CRC) are used for error detection (a single 32-bit CRC is more complex to calculate and provides only marginally better coverage). The link CRC (LCRC) is calculated on a link basis and covers both the data and control information except itself. The LCRC is initialized to ones for every micropacket. An end-to-end CRC (ECRC) is created by the originating source that covers all the data bytes of a message. The ECRC is initialized at the start of a message and thus provides a running CRC for the message. The LCRC is a common even term CRC used in disk devices. The ECRC is an odd term check function and thus catches many errors missed by the even term LCRC.

A transmission sequence (TSEQ) value is sent with each micropacket and is returned as a receive sequence (RSEQ) value for correctly received micropackets in the reverse direction control information from the other side of the link. The sequencing engine compares TSEQ to RSEQ values and clears the retransmission buffer and resets timers up to the acknowledged RSEQ value. Timer values are set for each TSEQ value that will cause a retransmission in the event of a missing RSEQ. The timer value is programmable allowing the user to optimize the retransmission scheme based on round trip latency and host sequence-processing times.

2.2 Virtual Channels and Flow Control

HIPPI-6400 uses four 256 micropacket-sized buffers that operate using credit-based flow control. The protocol requires that each VC may have only one message in progress at a time and specifies different message sizes for each VC. The virtual channels are assigned as: VC0 for control messages (≤ 2080 bytes), VC1 and VC2 for IP sized transmission (< 128 Kbytes), and VC3 must used the scheduled transfer mechanism (see Section 3) - using messages up to 4 gigabytes. After a Reset or Initialize operation, the source engine will acquire free buffer information from the destination engine and begin sending credits to the other end-point's source engine. Because the flow control mechanism sits outside the sequencing mechanism, the protocol will never credit buffers that were in error.

3. SCHEDULED TRANSFERS

One large advantage that HIPPI-6400 offers is a standardized scheduled transfer operation set capable of full data rate transmissions between the end devices. The scheduled transfer mechanism provides a handshake that allows the two devices to agree on transmission parameters including maximum transmission unit, total transmission size, buffer sizes, and buffer offsets. For administration and possibly security measures, 64-bits of key and transfer identification as well as port information are exchanged in the initial handshake. The HIPPI-6400 scheduled transfer aligns the transfer data into contiguously indexed buffers on the destination device. The destination device will allocate required buffers at startup allowing the actual data transfer to bypass an interrupt or polled upper layer controller.

3.1 Scheduled transfer parameters

<table>
<tr><td>Op</td><td>M-count</td></tr>
<tr><td>D-port</td><td>R-port</td></tr>
<tr><td colspan="2">Key</td></tr>
<tr><td colspan="2">T-id</td></tr>
<tr><td colspan="2">Bufx</td></tr>
<tr><td colspan="2">Offset</td></tr>
<tr><td colspan="2">T-len</td></tr>
<tr><td colspan="2">B-num</td></tr>
</table>

Figure 2: Scheduled transfer parameters

Figure 2 shows the parameters used in every scheduled transfer operation. A header micropacket that conveys routing information, ULA addresses, message length and translation flags begin all HIPPI-6400 messages, but scheduled transfer operations include a scheduled transfer parameter micropacket after the header. The top four fields are 16-bits each and the remaining fields are each 32-bits. The Op field carries the operational code. All control operations use virtual channel 0. The data operations which may use VC1, VC2, or VC3. The D-port (destination port), R-port (reply port), and Key values are used to authenticate a virtual connection during port setup. The virtual connection specifies a logical connection between end points which communicates all initial state information. The T-id (transfer identification) validates all control messages for a specific scheduled transfer on a virtual connection. The T-len (transfer length) declares the transfer size. B-num (block number) denotes the block within the transfer. M-count (message count), Bufx (buffer index), and Offset setup buffer tiling and later specify the buffer destination for transmitted data that was previously allocated. The buffer tiling concentrates required complexity on the source end point which yields better transfer efficiency.

3.2 Scheduled transfer hierarchy

Legacy HIPPI might stream a “jumbogram” until completion (a sometimes desired, deterministic characteristic) preventing other connections from communicating. The HIPPI-6400 standards body created a hierarchical data structure that allows better bandwidth sharing, between different sized messages and even between jumbograms. Figure 3 displays the scheduled transfer hierarchy.

The message at the bottom of this hierarchy is the only element that can hog the interface, but the inclusion of four virtual channels means that a large transfer will never block control messages on VC0 or IP communication across VC1 and VC2. The maximum message size is dictated by the smaller of the buffer sizes indicated by both end-points.

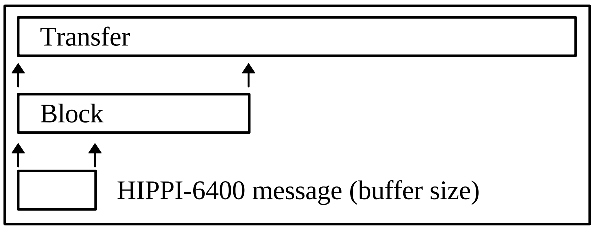


Figure 3: Scheduled transfer hierarchy

The block is used for striping or pipelining and is the upper layer retransmission and possibly buffer looping-quantity. Scheduled transfers use a request-to-send (RTS), clear-to-send (CTS) paradigm which is performed on a block basis. The destination end-point selects the block size based on an integral number of maximally sized messages.

Finally the transfer size indicates the size of the transmission (a special response indicates unlimited transmission size). The transfer size is chosen by the source end-point in a RTS operation (put), and by the destination end-point in a request-to-receive (RTR) operation (get).

3.3 Scheduled transfer example

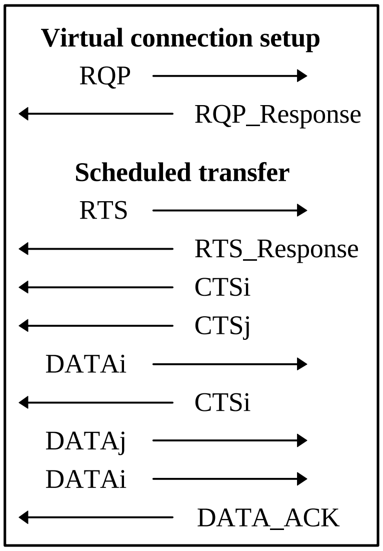


Figure 4: Scheduled transfer example

Figure 4 shows an example scheduled transfer exchange. The virtual connection is setup by a request-port (RQP) operation in which ports, keys and buffer sizes are exchanged. The port setup accommodates scheduled transfers by exchanging initial state information.

The scheduled transfer is initiated by an RTS and RTS_Response wherein a transfer identification, transfer size, and block size are communicated. The destination end-point allocates buffers for the transfer and may send clear-to-send (CTS) operations for as many blocks as can be accepted. In this example, the first DATA operation contains an error and a second CTS is sent asking for retransmission. Finally, both blocks arrive and the destination end-point sends a data acknowledgment.

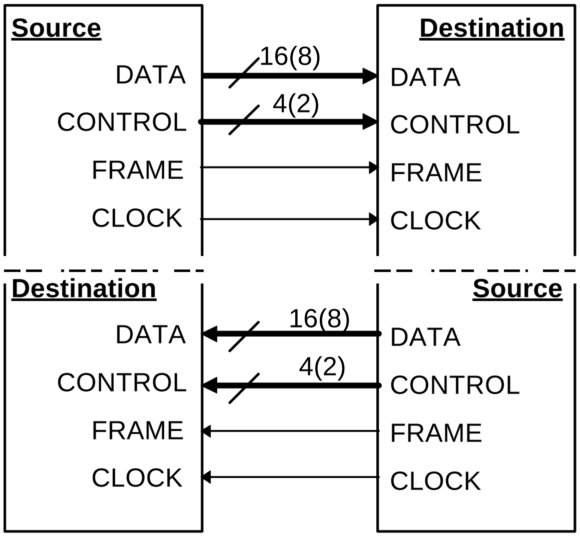
Reviewing the scheduled transfer in depth reveals the capability to perform operating system bypass transfers directly to DMA firmware (that interpret data operations and place data into allocated buffers). An OS Bypass mechanism is not needed for small transfers, but as the transfer size grows the operating system will be continually interrupted and eventually bottleneck the transfer. Dedicated I/O processors can sometimes account for some of this bottleneck, but the scheduled transfer mechanism is necessary for seamless high throughput transfers. The author strongly believes that other network infrastructures that require high bandwidth to the desktop should adopt a scheduled transfer mechanism like the one detailed in HIPPI-6400 (or possibly the same scheduled transfer mechanism in HIPPI-6400 for interoperability across translators).

4. SIGNALING INTERFACE

The HIPPI-6400 signaling interface contains four components: 4b/5b balanced line coding; training pattern and deskew logic; transceiver / receiver (TX/RX) pairs; and the cabling. Figure 5 (also from HIPPI-6400-PH) shows the signal lines for a single link. The frame signal stays positive for the first half of a micropacket and negative for the second half denoting the micropacket framing. All data and control lines are latched on both edges of the clock signal. A source synchronous clock design is used to complement the dynamic deskew method, i.e., a phase locked loop design would not allow frame alignment of the clock with the current training sequence and increases jitter effects.

The 4b/5b coding uses a similar idea as the Hewlett-Packard 20/24b encoding used in HIPPI-Serial and the H-P G-Link chip. A running disparity counter determines whether each 4-bit pattern should be inverted. The fifth bit (which is placed in the middle) tells the decode logic whether the pattern was inverted. A maximum run length of 11 may result but this proves to be inside the bandpass range for both optical and copper transceiver / receiver pairs.

Running a 10 gigabit transceiver is not presently cost efficient, so HIPPI-6400 specifies a 16-data-bit wide interface (500 MHz) for copper and an 8-bit wide interface (1 GHz) for fiber. The parallel copper and fiber interfaces require a line-to-line skew adjustment on the receiver side up to 10 ns to edge and frame align the data. The dynamic deskew circuitry requires a periodic (every 10us) training sequence to account for clock drift and environmental effects. The first part of the training sequence balances the disparity on each line. Next all lines go low for 14 ns to flush the deskew delay lines. Finally all lines invert and hold at one (for 14 ns) where the deskew circuitry detects and aligns both the frame and edge of all signals.



(Numbers in parenthesis are for an 8-bit system)

Figure 5: Signal lines

HIPPI-6400 has borrowed the experience of the Fibre Channel optical working group (comprised of more than ten parallel optics vendors) in standardizing the high performance interface. The group has decided to choose a connector at the upcoming December '96 ANSI meeting after vendor presentations in October '96. The basic high performance optical interface requires twelve lines per cable (one cable per direction) each signaling at 1 GHz. In order to traverse a 250 meter distance, conventional vertical cavity lasers may exceed eye safety requirements (IEC 82.5). Possible solutions to the eye safety problem (being considered by each vendor) are: longer wavelengths, shorter distance requirement, open fiber control, shutters, or a higher launch NA (~.275). Current worst case fiber-to-fiber skew is set at 7 ns (including TX/RX pair). As with any conventional parallel optics, multi-mode fiber will be used.

The lower cost copper interface (reach-challenged) may achieve 50 meter distances depending on whether equalization is incorporated. Unlike legacy HIPPI which could run in simplex mode, HIPPI-6400 must have a duplex connection and therefore a single cable has been selected with 100 differential pairs giving 50 signal lines when 44 are required (extra lines may be used for an Interconnect signal and a power line for an outboard optical extender.) Los Alamos National Laboratory created a 500 MHz signaling engine to test eye patterns on vendor cables and to analyze possible 4b/5b jitter problems suggested by Al Widmer of IBM. Current tests show no reason to abandon the simple (especially across 20 lines) 4b/5b coding scheme. The standards group is looking into using the Gore quad-ax for lengthy runs (~50 m), and other vendor cables for shorter runs. IBM and 3M will be presenting their Jitney solution for HIPPI-6400 at the September meeting as a possible replacement for copper. Jitney offers short optical runs at a cost comparable to copper using thicker optics for easier alignment.

5. SWITCHING

HIPPI-6400 uses non-blocking switches to preserve full bandwidth communications. Rings/loops were discussed as a low cost option to switching but rejected due to latency, bandwidth reduction, and implementation concerns. The HIPPI-6400-SC (switch control) standard also contains annexes explaining bridging, routing, and switching aspects in HIPPI-6400.

5.1 Switching requirements

The HIPPI-6400-SC standard specifies switching requirements including fair message interleaving, fair micropacket interleaving, error checking, error conditions, routing, independent input port address mapping, switch management and congestion management. HIPPI-6400 switches use a 16-bit logical address for all switching in the HIPPI-6400 domain. HIPPI Media Access Controller (MAC) Headers contain the logical address for the message as well as a 64-bit network address. The logical address specifies a single route through the fabric to the destination device. The ECRC (which should only be generated by the originating end-point) ensures that no part of the HIPPI-6400 Message (other than control sideband bits) changes through the fabric.

5.2 Network Administration

Administration is performed on HIPPI-6400 networks by using an Admin type micropacket (as indicated in the TYPE field). All Admin micropackets consist of a single micropacket and use VC1 for requests and VC2 for responses (simple acknowledgment flow control mechanism.) Switches and host controllers take Admin micropackets out of the data stream and process the requested command. As Admin micropackets must be handled by a processing system, switches may discard Admin micropackets when Admin micropacket buffers overflows. Admin micropackets perform many functions including: switch and end-host bootstrapping/auto-configuration, providing logical addresses to end-points, reconfiguring switch routing tables, ARP, RARP, and an optimized subset of simple network management protocol (SNMP). In-band switch management will provide fast configuration, a small level of security, and a service that legacy HIPPI users have long awaited. The standards group is currently working on ironing out details of switch configuration and defining the administration operations set. Extenuating circumstances may require prototype switches to implement a subset of the in-band administration (may exclude switch table configuration and ARP/RARP).

6. CURRENT WORK

6.1 Legacy HIPPI support

HIPPI protocols such as framing protocol (FP) and link encapsulation (LE) will run over HIPPI-6400 in the same fashion as they do now. The group will add the scheduled transfer mechanism to the HIPPI-800 capability as well as defining HIPPI multi-path (HIPPI-MP) using the striping capabilities of the scheduled transfer. A translator has been in development since the standard started that will aggregate multiple legacy HIPPI links onto a HIPPI-6400 link at full rate. Los Alamos National Laboratory, who has one of the largest HIPPI networks, is dedicated to preserving legacy HIPPI in the new standard without requiring any sort of “interoperability upgrade”.

6.2 Current work

Two very large programs are driving the HIPPI-6400 development. The Los Alamos and Livermore National Laboratories Accelerated Strategic Computing Initiative (ASCI) program requires phenomenal computing power, storage, and visualization for 3d fluid flow simulations. Raytheon/E-Systems is contracted to provide a similar system on a tighter time budget. Silicon Graphics has contracted to both groups to build end-host adapters that support close to full rate (700 Mbytes/s) HIPPI-6400 transfers. Raytheon/E-Systems is building a 32 port, low-latency HIPPI-6400 switch. Essential Communications, a leader in HIPPI products, is designing a HIPPI-800 translator for HIPPI-6400. Los Alamos National Laboratory in cooperation with Optivision is developing a multi-platform tester that accepts HIPPI 800/1600, HIPPI-6400, and high speed ATM/SONET interface boards. All the above hardware projects started development at least 6 months ago and all are slated for success.

6.3 HIPPI-6400 Simulation Modeling

The author has been simulating various aspects of the HIPPI-6400 specification individually, but is concurrently developing a modular C++ model to investigate the sequencing and flow control mechanisms when errors occur. If time permits, similar models will be constructed for Fibre Channel, legacy HIPPI, and Scalable Coherent Interface (SCI).

HIPPI-6400 can be broken into a channel hierarchy. The inner “link” module handles data delivery, error checking, and retransmission. The outer “virtual channel” module manages link-to-link flow-control to stop a Source from overrunning the buffers in a Destination. Just as the virtual channel communicates through the link channel, the link channel transmits actual data through a signaling channel. The actual application talks to the virtual channel through a messaging channel. In Figure 6 below, these modules are shown as they correspond to the ANSI documentation and are called “channels”.

<table>
<tr><td rowspan="2">Annexes</td><td>HIPPI</td><td>IPI-3</td><td>FC</td><td>IP</td><td>ATM</td><td>802.2</td></tr>
<tr><td colspan="3">Connection Message Handling</td><td colspan="3">Legacy Services</td></tr>
<tr><td colspan="4">PH/SC</td><td colspan="2">Messaging Channel</td><td></td></tr>
<tr><td colspan="4" rowspan="3">PH</td><td colspan="2">Virtual Channel</td><td></td></tr>
<tr><td colspan="2">Link Channel</td><td></td></tr>
<tr><td colspan="2">Signaling Channel</td><td></td></tr>
</table>

Figure 6: Channel Hierarchy

As shown in Figure 6, there are four levels in the HIPPI-6400 specification: messaging channel, virtual channel, link channel, and signaling channel. The top messaging channel is still being discussed in the ANSI working group and the messaging channel may not be worth modeling unless it presents unexpected bottlenecks. Figure 7 displays the various channels and functions they encompass.

Although bandwidth is an important aspect of every network, throughput is even more important. HIPPI-6400 concentrates on removing latency from the network pipe by using small micropackets and moving sluggish software operations into the hardware levels. Another aspect is continuous micropacket transmission. Previous HIPPI physical layers (HIPPI-800 and 1600), have a connection setup interval prior to allowing actual data flow. This requires an added round trip latency for each message that one sends across the network. When no data micropackets are being sent, HIPPI-6400 transmits various link management micropackets, (e.g., credit-only micropackets, retraining sequences, Admin micropackets, or when there is nothing else to send, null micropackets).

Messaging Channel	Virtual Channel	Link Channel	Signaling Channel
OS Bypass	VC Control	Link Control	Signaling Control
Header Creation	VC Buffering	LCRC	Bit Ordering
Switching	Credit-Accounting	I/O Buffering	Deskew
Administration Messaging	ECRC (also part of Link Layer)	Sequence Number Handling	Retraining Sequence (Overrun Prevention, every 10 us)
	Credit-Only micropackets	Retransmission	Balanced Coding
		ECRC (single)	Signaling Interface
		Null micropackets	

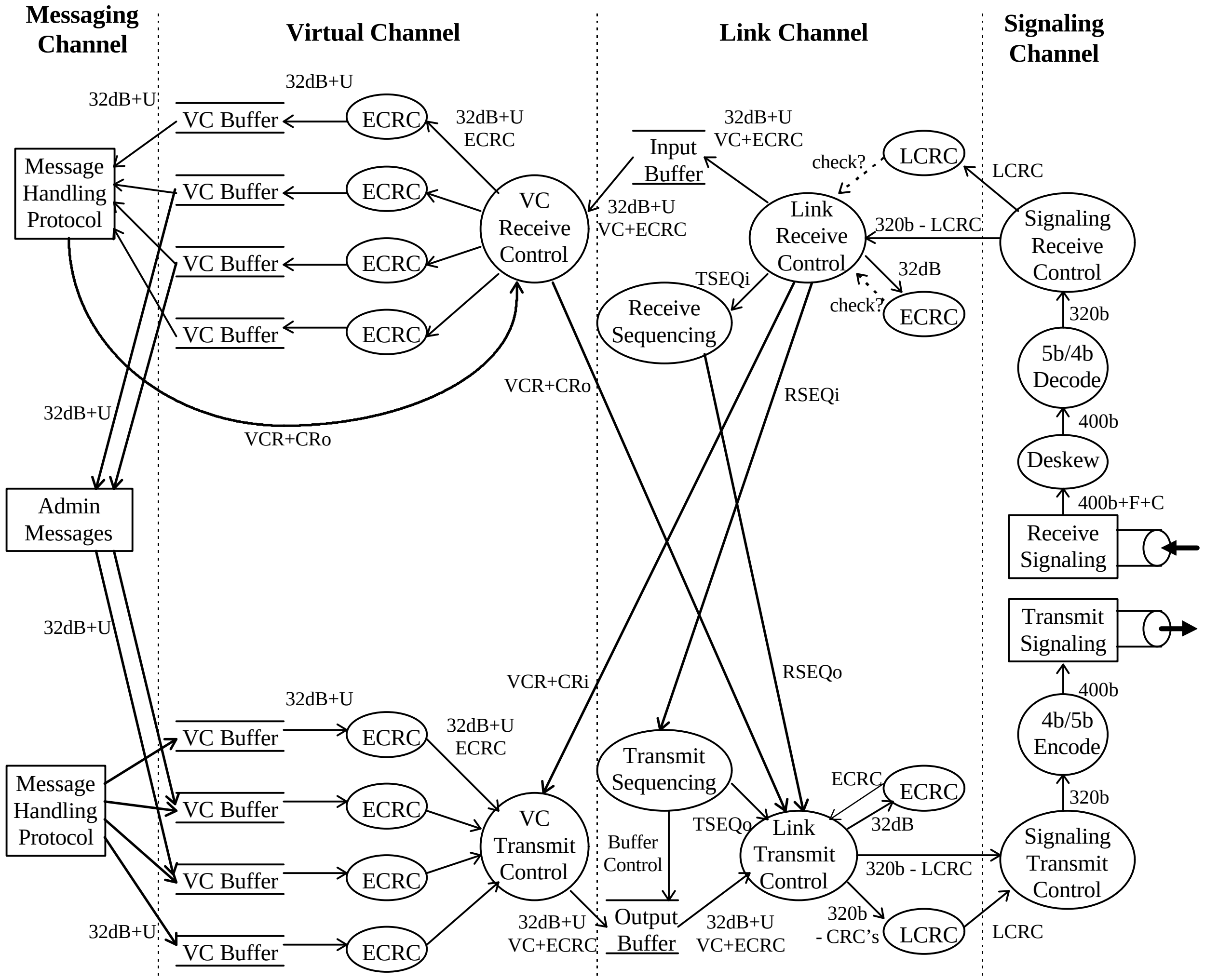
Figure 7: Channel Functions

Despite the tremendous rate, (i.e., one thousand times Ethernet speed), that HIPPI-6400 transmits at, many signals must communicate between logical units to transmit a single micropacket. Figure 8 (next page) shows the basic data-flow model that is being implemented in code. The model does not contain any state information, only directional signal communication. The model is broken into four channels and both the Source and Destination of a single node are shown. The model is the author’s interpretation of the HIPPI-6400-PH standard.

6.4 Conclusions, where did HIPPI-6400 come from?

HIPPI-800 was conceived in the mid 80’s as a large data pipe for simple rapid data streaming by engineers at Los Alamos. HIPPI-6400 retains many of the legacy HIPPI successes: simplicity and excellent flow control. HIPPI-6400 also borrows ideas from multiple forums such as: small micropacket size (low latency) and virtual channels (multiplexing capability) from ATM; parallel fiber interface and memory paradigm from Scalable Coherent Interface (SCI); and standardized auto-configuration and network management as requested by multiple legacy HIPPI users. HIPPI-6400 is intended as a digital LAN backbone of the near future, as a direct interconnect for today’s power-stations (tomorrow’s workstations) and finally as a technological advancement to push the realizable capability of the computer interconnect market. Please note that this report cannot convey intricate details of the standard and is only an introduction. See the actual HIPPI-6400-PH and HIPPI-6400-SC standards for implementation details.

As shown there are many aspects to HIPPI-6400 and the ANSI working group tries to choose the best system, placing key, implementation, and performance tradeoffs in tunable parameters. The veteran design group uses past network specification experience and simulated results to choose the constituent parts of HIPPI-6400. Truly, some design choices are disputable and optional choices may be preferable to certain consumers. Unfortunately, Fibre Channel has shown that allowing options in a networking scheme leads towards differing incompatible implementations. One of the chief duties of an interconnect standard is to promote modular interoperability between network nodes.



Notes:	
32dB: 32 data Bytes	**U**: Upper Level Control (TYPE, TAIL, ERROR)
320b: 320 bit unencoded micropacket	**400b**: 400 bit encoded micropacket
check?: boolean CRC result	**F+C**: Frame and Clock signals
X**i**: incoming	X**o**: Outgoing

Figure 8: Data flow model

6.5 Further reading

ANSI X3.xxx - 199x, High-Performance Parallel Interface – 6400 Mbit/s Physical Layer (HIPPI-6400-PH)

ANSI X3.xxx - 199x, High-Performance Parallel Interface – 6400 Mbit/s Switch Control (HIPPI-6400-SC)

ANSI X3.183 - 1991, High-Performance Parallel Interface – Mechanical, Electrical, and Signaling Protocol Specification (HIPPI-PH)

ANSI X3.xxx - 199x, HIPPI Serial Specification (HIPPI-Serial)

HIPPI-6400 Web Page: http://www.cic-5.lanl.gov/~jamesh/hippi64/
HIPPI Networking Forum Web Page: http://www.esscom.com/hnf/

HIPPI Documents: http://www.cic-5.lanl.gov/~det/

7. ACKNOWLEDGMENTS

All information contained in this document came from the ANSI X3T11 HIPPI-6400 working group which encourages participation in the development and design of current and future network technologies. The author thanks all attending members for their insights and ideas (even the divergent ones). Specifically, kudos to Don Tolmie, the father of HIPPI, for his continued persistence in pushing the standard development, keeping the group honest, and performing the majority of word-smithing the next generation networking document. The author has been paid as a University of Arizona graduate research assistant under Dr. Indra Widjaja by Los Alamos National Laboratory. Thanks also go to Dr. Widjaja for putting up with the transport protocol junkie (the author) and for his guidance.

The Los Alamos National Laboratory is operated by the University of California for the United States Department of Energy under contract W-7405-ENG-36. The author's work was performed under auspices of the U. S. Department of Energy. This document is unclassified (LAUR96-3033).